Software News and Update

PSI3: An Open-Source *Ab Initio* Electronic Structure Package

T. DANIEL CRAWFORD,[1] C. DAVID SHERRILL,[2] EDWARD F. VALEEV,[1] JUSTIN T. FERMANN,[3] ROLLIN A. KING,[4] MATTHEW L. LEININGER,[5] SHAWN T. BROWN,[6] CURTIS L. JANSSEN,[5] EDWARD T. SEIDL,[7] JOSEPH P. KENNY,[5] WESLEY D. ALLEN[8]

[1]*Department of Chemistry, Virginia Tech, Blacksburg, Virginia 24061*
[2]*Center for Computational Molecular Science and Technology, School of Chemistry and Biochemistry, Georgia Institute of Technology, Atlanta, Georgia 30332*
[3]*Department of Chemistry, University of Massachusetts, Amherst, Massachusetts 01002*
[4]*Department of Chemistry, Bethel University, St. Paul, Minnesota 55112*
[5]*Sandia National Laboratories, Livermore, California 94551*
[6]*Pittsburgh Supercomputing Center, Pittsburgh, Pennsylvania 15213*
[7]*Lawrence Livermore National Laboratories, Livermore, California 94550*
[8]*Center for Computational Chemistry and Department of Chemistry, University of Georgia, Athens, Georgia 30602*

Received 21 August 2006; Accepted 1 October 2006
DOI 10.1002/jcc.20573
Published online in Wiley InterScience (www.interscience.wiley.com).

Abstract: PSI3 is a program system and development platform for *ab initio* molecular electronic structure computations. The package includes mature programming interfaces for parsing user input, accessing commonly used data such as basis-set information or molecular orbital coefficients, and retrieving and storing binary data (with no software limitations on file sizes or file-system-sizes), especially multi-index quantities such as electron repulsion integrals. This platform is useful for the rapid implementation of both standard quantum chemical methods, as well as the development of new models. Features that have already been implemented include Hartree-Fock, multiconfigurational self-consistent-field, second-order Møller-Plesset perturbation theory, coupled cluster, and configuration interaction wave functions. Distinctive capabilities include the ability to employ Gaussian basis functions with arbitrary angular momentum levels; linear R12 second-order perturbation theory; coupled cluster frequency-dependent response properties, including dipole polarizabilities and optical rotation; and diagonal Born-Oppenheimer corrections with correlated wave functions. This article describes the programming infrastructure and main features of the package. PSI3 is available free of charge through the open-source, GNU General Public License.

J Comput Chem 00: 000–000, 2007

Key words: quantum chemistry; electronic structure; coupled cluster theory; multireference configuration interaction; computational chemistry

Introduction

Ab initio quantum chemical programs are exceedingly complex, often consisting of hundreds of thousands to even millions of lines of computer code, making the development of new methods an arduous task. One way to combat this complexity is to construct a suite of robust software tools for carrying out tasks common to most electronic structure models. In 1999, we began development of "PSI3"—a new electronic structure package with an extensive infrastructure designed to speed the implementation of new theories and to remove common bottlenecks. PSI3 is written entirely in the C and C++ programming languages for greater ease of maintenance

Correspondence to: T. D. Crawford; e-mail: crawdad@vt.edu; C. D. Sherrill; e-mail: sherrill@chemistry.gatech.edu; E. F. Valeev; e-mail: valeev@vt.edu

Contract/grant sponsor: National Science Foundation; contract/grant numbers: CHE-0133174 (T.D.C.), CHE-0094088 (C.D.S.)

Contract/grant sponsor: Sandia National Laboratories Lab Directed Research and Development; Contract/grant numbers: 25580 (T.D.C.), 24732 (C.D.S., E.F.V.)

Contract/grant sponsors: IBM SUR (Georgia Tech Center for Computational Molecular Science and Technology); Camille and Henry Dreyfus Foundation New Faculty Awards (T.D.C., C.D.S.); Research Corporation Cottrell Scholar Award (T.D.C.)

and portability, and includes new I/O libraries that have no practical file size or file system-size limitations, as well as an improved checkpoint file structure and greater automation of typical tasks.

This new infrastructure has enabled the rapid implementation of a series of new integral, coupled cluster, and configuration interaction (CI) codes. The current package has the capability to determine wave functions and energies based on a variety of theories, including spin-restricted and -unrestricted Hartree-Fock (RHF, UHF, and ROHF); CI, including a variety of multireference CIs and full CI; coupled cluster, including coupled cluster singles and doubles (CCSD) and CCSD with both noniterative [CCSD(T)] and iterative triples (CC3); second-order Møller-Plesset perturbation theory (MPPT), including explicitly correlated second-order Møller-Plesset energies (MP2-R12); and complete-active-space self-consistent field (CASSCF) theory. In addition, for many of these theories, analytic energy gradients and various molecular properties are also available.

The development of PSI3 has greatly benefitted from the algorithmic advances of the package's predecessors. The PSI programs originated in the group of Prof. Henry F. Schaefer of the University of Georgia's Center for Computational Chemistry. Some of the earliest contributions to the program included a direct CI code (written in 1976 by Robert Lucchese, now at Texas A&M), the well-known graphical unitary group CI (GUGACI) program (written in 1977–78 by Bernard Brooks, now at the National Institutes of Health), and a Gaussian integral code (written in 1978 by Prof. Russell Pitzer, at Ohio State University). From 1978–1987, the package was known as the BERKELEY suite, and after the Schaefer group moved to the University of Georgia, it was renamed PSI. Thanks primarily to the efforts of Drs. Curt Janssen (now at Sandia National Laboratory, Livermore) and Ed Seidl (now at Lawrence Livermore National Laboratory), the package was ported to UNIX systems, and substantially improved with new input formats and a C-based I/O system. Although PSI3 contains little of the previous code, many of the algorithmic techniques of earlier versions were incorporated into its present design.

This paper describes the capabilities and programming infrastructure of PSI3. The package offers developers of quantum chemistry programs a convenient, easily maintained programming environment with a relatively shallow learning curve for the construction of new methods, and we welcome new contributions. In order to provide the widest distribution and greatest ease of access to both developers and users, we distribute the programs under the open-source, GNU General Public License (See www.gnu.org for a complete explanation of this license.) The most recent version of the program (version 3.2 as of this writing) may be obtained at `http://www.psicode.org/`. Up-to-date user and programmer manuals are also available at this website.

PSI3 Development Tools

Input Parsing

In order to provide a flexible and simple free-format user input, PSI3 includes an input-parsing library similar to that found in many programming-language compilers. The basic PSI3 input grammar is provided by low-level parsing and lexical analysis functions (constructed using the standard UNIX utilities `yacc/bison` and `lex/flex`). These functions identify the various keywords and their associated values and store them in a hierarchical data structure (a tree) for later use by the programmer via a series of high-level functions. A major advantage of the PSI3 input grammar is its flexibility: the input-parsing library makes it possible to input very simple data, such as single integer or boolean types, or very complicated structures, such as multidimensional arrays containing multiple types, such as integers, strings, floating-point numbers, or even other arrays. The hierarchical nature of the keyword tree allows one to overload keywords easily. For example, using the same keyword `convergence`, placed in different sections of the input file, users can specify one convergence level for the SCF and coupled cluster ground-state energy calculations, but a different level for subsequent excited-state calculations.

Input/Output

As is typical for quantum chemistry program packages, PSI3 makes extensive use of raw binary (also called "direct-access") files, which, depending on the type and size of the calculation, may range in size from a few kilobytes to several hundred gigabytes. Rather than using low-level C or Fortran functions alone, PSI3 uses a flexible, fast I/O system that gives the programmer and user significant control over the organization and storage of data. Some of the features of the PSI I/O system include:

- A user-defined disk striping system in which a single binary file may be split across several physical or logical disks.
- A file-specific table of contents (TOC) that contains starting and ending addresses for each data item.
- An entry-relative page/offset addressing scheme that avoids file-global file pointers, which can limit file sizes.

The PSI3 disk-striping system may be thought of as a simplified redundant array of independent disks (RAID) approach that is controllable at the user level. Although many modern computer systems make use of fast hardware-based RAID, other common computing environments, such as community-shared servers at universities and national laboratories, may provide several independent scratch disk areas. The PSI3 user can specify explicitly where certain types of files are to be stored or how classes of files are to be distributed across multiple disks/directories.

The TOC structure of PSI3 binary files provides several advantages over older I/O systems. First, the ordering of entries in the file is inconsequential, and the programmer thus needs no "map" of the binary file to access data. For example, data items in the TOC are identified by keyword strings (e.g., "`Nuclear Repulsion Energy`") and the global address of an entry is known only to the TOC itself, and never to the programmer. Hence, if the programmer wishes to read or write an entire TOC entry, he/she need only provide the TOC keyword and the entry size (in bytes) to obtain the data. Second, the TOC makes it possible to read only pieces of TOC entries (say a single buffer of a large list of two-electron integrals) by providing the appropriate TOC keyword, a size, and a starting address relative to the beginning of the TOC entry. In short, the TOC design hides all information about the global structure of the direct access file from the programmer and allows him/her to be concerned only with the structure of individual entries.

Finally, data are addressed in PSI3 using a page/offset scheme rather than a single bytewise file pointer. Thus the direct-access file itself is viewed by the I/O system as a series of pages, each of which contains an identical number of bytes. The global address of the beginning of a given entry is stored on the TOC as a page/offset pair comprised of the starting page and byte-offset on that page where the data reside. The entry-relative page/offset addresses which the programmer must provide work in exactly the same manner, but the 0/0 position is taken to be the beginning of the TOC entry rather than the beginning of the file. This frees the programmer from concerns about global file structure, and allows essentially unlimited binary file sizes.

The Checkpoint File

PSI3 makes use of a checkpoint file, which contains the most frequently used information about the current computation, including the molecular geometry, the basis set(s), the molecular orbital coefficients, symmetry information, etc. The data are stored using the TOC format described in the previous section, and the programming interface to the file is handled exclusively through a specialized library. The checkpoint functions are designed for simplicity, and thus allow the programmer to retrieve data structures almost trivially, regardless of their complexity. (For example, the molecular orbital coefficients or a density matrix may be obtained with only a single line of code.) Similarly, straightforward functions exist for writing data to the checkpoint file, and new entries may be easily added to the library. In addition, the checkpoint file may contain sections that store similar information that must be kept separate (e.g., data for each displacement in a finite-difference gradient calculation, or basis-set information for composite calculations, such as the Gaussian-*n* approaches). Together, the I/O and checkpoint libraries have dramatically simplified the program development. In previous versions of PSI, for example, it was extremely difficult to extend the maximum angular momentum for basis functions, because the checkpoint file had arrays of fixed length, and all programs reading the checkpoint file would have to be modified. Such concerns are no longer an issue—and moreover, PSI3 can handle basis functions of arbitrary angular momentum using the libraries described in the next section.

Integral Evaluation Libraries

Every calculation in PSI3 involves evaluation of numerous matrix elements (integrals) over atom-centered Gaussian functions. This task is carried out in PSI3 by a library of functions called `LIBINT`, which is capable of computing not only electron repulsion integrals (ERIs), by far the most common type of integral in electronic structure theory, but also first and second derivatives of ERIs with respect to nuclear coordinates, and various integrals appearing in linear-R12 theories.[1,2]

`LIBINT` uses recursive schemes that originate in the venerated Obara–Saika method[3] and Head-Gordon and Pople's variation thereof.[4] The basic problem with the Obara–Saika recurrence relations is that their naïve implementation results in inefficient recomputation of numerous intermediate integrals. One could keep track of which integrals have been computed already, and thus avoid needless recomputation, but such a scheme would require an excessive number of conditional statements in the code and would thus still be very inefficient. `LIBINT` avoids these difficulties by solving the recursion problem in advance, such that the critical (machine-generated) low-level code for computing individual quartets of integrals contains no conditional statements whatsoever.

The program that generates the code is referred to as the optimizing compiler. The resulting code executes very efficiently on superscalar architectures, e.g. critical pieces of `LIBINT` run at >50% of the peak FLOP rate on popular x86 and AMD64 processors. These high efficiencies are achieved without the use of vectorization techniques, which may increase the memory footprint and complicate programming. Another advantage is that new recurrence relations are easy to implement efficiently. For example, a library for computing the integrals required for linear R12 theories was implemented in mere weeks, instead of the months expected for most other program packages.

The programmer may access `LIBINT`'s machine-generated code (The `LIBINT` interface is described in detail at http://www.chem.vt.edu/chem-dept/valeev/libint) via an external interface with two key features:[5]

- Integral evaluation is performed using an object-oriented design with C-style data-structures containing the key parameters, which must be provided by the calling routine.
- The code that computes the integrals is reentrant, and therefore multiple integral evaluation objects can coexist and be used at the same time. Thus, `LIBINT` is suitable for multithreaded execution.

A tradeoff of the code-generation approach is that the amount of source code—and therefore the size of the resulting executable program—can grow dramatically with the maximum orbital quantum number. For example, the source code can grow to 30 megabytes or more if the library supports up to $l = 7$ (*k*-type functions). However, this code bloat does not currently seem to be a problem in practice, thanks to effective instruction caching on modern scalar processors. Also, electronic structure calculations with basis functions of orbital quantum numbers above $l = 7$ are of little practical value,[6] thanks to the advances in explicitly correlated methodologies.[7]

Manipulation of Multi-index Quantities

PSI3 contains two specialized libraries supporting I/O operations on files containing two- and four-index quantities (molecular integrals, density matrices, wave function amplitudes, intermediates, etc.). The first, referred to as the "integrals with labels" (IWL) library, is particularly convenient dealing with sparse data lists (e.g., the atomic-orbital integral list for a large molecule). In the IWL format, integrals are stored along with their orbital labels in a file partitioned into multiple buffers. Word sizes for the orbital labels and for the values of the integrals are defined with the C-language `typedef` directive, so that they may be changed very easily. The IWL library provides low-level functions to read/write a single buffer of integrals, which allows the development of robust out-of-core algorithms. The library handles all the bookkeeping details via a header at the start of each buffer of integrals; the header contains information such as the number of integrals in the current buffer, whether the current buffer is the last one on disk, etc. The library can also make use of permutational symmetries in the data as well as precision cutoffs, to avoid storing redundant or neglible integrals.

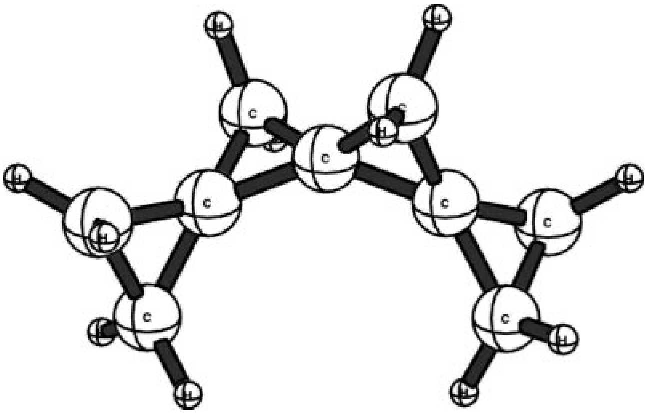

Figure 1. The C_2-symmetry structure of (*P*)-[4]triangulane (trispiro [2.0.0.2.1.1]nonane).[35]

The second library, referred to as the "direct-product decomposition" (DPD) library,[8] is designed for efficient storage and manipulation of quantities with Abelian point-group symmetry, especially in the context of methods such as many-body perturbation theory and coupled cluster theory. The storage approach in `libdpd` is fundamentally different from that of `libiwl` in that the data are stored in a "supermatrix" format, with separate arrays provided automatically by the library for translating orbital indices to row and column indices and *vice versa*. Permutational symmetry (or antisymmetry) can be handled automatically, and indices can be packed or unpacked on-the-fly during read/write operations. The supermatrices are symmetry-blocked to avoid storage of zeroes, and the library can seamlessly handle both totally symmetric quantitites (e.g., two-electron integrals or ground-state wave function amplitudes) or nontotally symmetric ones (e.g., perturbed or excited-state wave functions).

The DPD library also provides functions for evaluating binary products of multi-index quantities, such as those often found in many-body theories. These "contraction" functions are flexible in that they allow automatic transposition of input and output quantities and they may be applied to both two- and four-index arrays. In addition, the contraction functions automatically select efficient out-of-core algorithms in memory-limited calculations. Finally, `libdpd` provides built-in functions for sorting multi-index quantities to any desired ordering, and a memory cache designed to reduce I/O wait times.

PSI3 Capabilities

Coupled Cluster Methods

The coupled cluster approach is widely regarded as one of the most reliable quantum chemical methods because of its ability to provide exceptional accuracy in predictions of a wide variety of molecular properties, including geometrical parameters, vibrational transition frequencies and intensities, electronic excitation spectra, etc.[9–13] Development efforts in PSI3 have focused heavily on efficient coupled cluster programs, whose programming infrastructure relies on the DPD library described earlier. The most recent version of the package includes single-point energies at several levels, including CCSD,[14,15] CCSD plus a perturbative estimate of connected triples—the popular CCSD(T) model,[16,17] and an iterative triples model, CC3.[18,19] In addition, analytic energy gradients are available for CCSD with several types of reference orbitals.[20–23]

Electronic excitation energies may be computed using PSI3 at the CC2,[24] CCSD,[25] and CC3[19,26] levels, including RHF, ROHF, and UHF reference wave functions. Analytic energy gradients are also available for EOM-CCSD,[27] thus allowing structural optimizations on excited-state potential energy surfaces. Finally, we have recently implemented CC2- and CCSD-level linear response methods[28,29] for computing dipole polarizabilities[30] and optical rotation angles,[31–35] among others.

Numerous applications of the PSI3-coupled cluster programs have been reported recently. PSI3 was used, for example, in a recent study of the excited states of several radicals, such as the allyl radical (C_3H_5), including the first report of excitation-energy calculations of radicals involving iterative triple excitations.[26,36] The largest CCSD linear-response calculations carried out with the package include optical rotatory dispersion modeling of the σ-helicene, [4]triangulane (trispiro[2.0.0.2.1.1]nonane, Fig. 1), a C_2-symmetry species exhibiting exceptionally large specific rotations[35,37] (Table 1). More recently, we have carried out as-yet unpublished CCSD calculations of optical rotations for nonsymmetric molecules such as (1*R*,5*R*)-β-pinene ($C_{10}H_{16}$, 310 basis functions, aug-cc-pVDZ) and (*S*)-fluoroxirane (C_2H_3OF, 458 basis functions, aug-cc-pVQZ). The CCSD energy component of the latter calculation required only 17 hours (wall time) on a 2.2 GHz Opteron 248 single-CPU system.

The PSI3 coupled cluster codes have also served as a testbed for local correlation techniques, which are under active development by the authors.[40,41] Table 2 summarizes the coupled cluster features of the most recent version of the code.

Determinant-Based CI and CASSCF Methods

Simple CI singles and doubles (CISD) approach widely used in the 1970s and 1980s, has been superceded by the more robust (and size-extensive) CCSD method. For this reason, PSI3 focuses on more complete CI wave functions, including triple, quadruple, or higher-order substitutions, and we have developed a determinant-based program, DETCI,[42] which is very efficient for this purpose. For example, in a recent study[43] tight convergence of the full CI wavefunction for the CH_4 molecule in a 6-31G* basis (more than 566 million determinants with no orbitals frozen) required only two

Table 1. Theoretical and Experimental Values of the Optical Rotation (in deg dm^{-1} $(g/mL)^{-1}$) of (*P*)-[4]Triangulane (trispiro[2.0.0.2.1.1]nonane) as a Function of Wavelength.

Wavelength (nm)	CCSD/aug-cc-pVDZ	Expt.
589	196.0	192.7
578	204.5	201.3
546	232.9	229.7
436	398.7	400.2
365	635.4	648.2

Experimental values were taken from refs. 40 and 41.

Table 2. Coupled Cluster Methods Currently Available in PSI3.

Reference	Method	Energy	Gradient	Excited states	Linear response
RHF	CC2	Y	N	Y	Y
RHF	CCSD	Y	Y	Y[a]	Y
RHF	CCSD(T)	Y	N	–	–
RHF	CC3	Y	N	Y	N
ROHF	CC2	Y	N	Y	N
ROHF	CCSD	Y	Y	Y[a]	N
ROHF	CCSD(T)	Y	N	–	–
ROHF	CC3	Y	N	Y	N
UHF	CC2	Y	N	Y	N
UHF	CCSD	Y	Y	Y[a]	N
UHF	CCSD(T)	Y	N	–	–
UHF	CC3	Y	N	Y	N
Brueckner	CCD	Y	N	N	N
Brueckner	CCD(T)	Y	N	–	–

[a] Analytic gradients for excited states are also available.

days on a 3.2 GHz Intel Xeon EM64T workstation. PSI3 has been used for full CI computations with more than a billion determinants on personal computers. The ability to perform efficient full CI computations has been invaluable in assessing the quality of standard models for bond-breaking reactions,[44–47] determining the effect of higher-order correlation in spectroscopic constants,[48–50] and in testing a variety of new electronic structure methods, including optimized-orbital coupled cluster,[51,52] spin-flip approaches,[53,54] and hybrid coupled-cluster/perturbation theories.[43,55]

The PSI3 program can perform any CI truncated at a given substitution level, such as CISD, CI with singles, doubles, and triples (CISDT), etc., up to the full CI limit. It can also perform restricted active space (RAS) CI computations as introduced by Olsen et al.,[56] in which the CI space includes all determinants (of the proper symmetry) that conform to user-defined rules about how many electrons may be placed in each of several orbital subspaces. The CI components of PSI3 are written in C and C++, with the CI vectors represented as C++ objects. This allows the implementation of a wide variety of iterative diagonalization methods at an abstract level, without regard to the underlying memory structure of the vectors. Indeed, PSI3 implements a large number of diagonalization methods proposed in the literature, which we have compared in a systematic study[57] (see also refs. 58, 59).

PSI3 has also been extended to handle certain "class-selected" CI spaces[60] which choose determinants based on the occupancy of different orbital subspaces, but which do not quite fit the original RAS CI selection scheme. This has been accomplished by adding a fourth orbital subspace, which we call RAS IV, and allowing additional restrictions on allowed determinants. These capabilities have been used to investigate certain compact CI wave functions which include limited triple and quadruple substitutions.[61] They have also been used to implement a "spin complete" version of Krylov's "spin flip" approach[62–64] by adding those determinants required to form $\hat{S}^2$ eigenfunctions. At the lowest level of the theory (spin flip self-consistent-field) the spin complete version can reduce errors by as much as 2/3.[54] Another unique capability of PSI3 is that it can generate Møller-Plesset perturbation theory energies to an essentially arbitrary order, using the approach of Handy and coworkers.[65,66] This capability has been used to perform the first study of spectroscopic constants of small molecules to high orders in the MPn series (with n up to 167).[67]

Finally, PSI3 can carry out a number of multiconfiguration self-consistent-field (MCSCF) calculations. We have implemented a two-step MCSCF algorithm to obtain CASSCF or RAS self-consistent-field[68] wave functions. For the orbital/orbital Hessian, we have implemented the exact Hessian, an exact diagonal Hessian, and the approximate diagonal Hessian of Chaban et al.,[69] which works well in our experience. Rather large active spaces, such as 14 electrons in 14 orbitals, present no difficulties for the CASSCF code in PSI3.

Linear R12 Second-Order Møller-Plesset Theory

The highly accurate wave function methods described in the previous two sections often suffer from large absolute errors when used with typical basis sets due to the use of n-electron bases composed of one-electron products. Explicitly correlated methods are capable of recovering correlation energy more efficiently by using an n-electron basis that depends on the interelectronic distances r_{ij}. The so-called linear R12 methods originally developed by Kutzelnigg and others[1,2,7,70–72] reduce basis-set incompleteness errors and thus yield faster asymptotic basis set convergence rates than their conventional counterparts. This improved performance is due to the inclusion of terms linear in the interelectronic distance into the wave function. For example, the linear R12 *ansatz* for the first-order wave function includes both the standard doubly excited determinants plus the terms linear in r_{ij}:

$$|\psi^{(1)}\rangle = \sum_{i<j,a<b} d^{ij}_{ab}|ab\rangle\langle ij|\psi^{(0)}\rangle + \sum_{i<j,k<l} c^{ij}_{kl}\hat{Q}_{12}|kl\rangle r_{12}\langle ij|\psi^{(0)}\rangle, \qquad (1)$$

where $\hat{Q}_{12}$ is a two-particle projector that ensures the orthogonality to the conventional double substitutions, and d^{ij}_{ab} and c^{ij}_{kl} are wave function coefficients. The operators $|ab\rangle\langle ij|$ and $|kl\rangle r_{12}\langle ij|$ in eq. (1) produce doubly substituted determinants and antisymmetrized r_{12}-multiplied orbital products, respectively. The latter terms produce

Table 3. Basis Set Convergence of Hartree-Fock and Valence Second-Order Electronic Energies (in E_h) for Acrylonitrile.

Basis set	RHF	MP2	MP2-R12/A
cc-pVDZ	−169.781 073	−0.540 316	–
cc-pVTZ	−169.825 322	−0.657 043	–
cc-pVQZ	−169.836 313	−0.698 754	–
cc-pV5Z	−169.838 739	−0.715 007	–
cc-pV6Z	−169.839 056	−0.722 457	–
"19s2i"	−169.839 101	−0.724 315	−0.733 680

The correlation energy converges much more slowly to the basis set limit than the Hartree-Fock energy. The MP2-R12 energy is significantly closer to the basis set limit than the conventional MP2 energy.
Explicit basis-set designation: 19s14p8d6f4g4h2i.

Table 4. Diagonal Born-Oppenheimer Correction for $H_2{}^{16}O$ Computed with aug-cc-pVDZ Basis Set at Several Levels of Theory

Wave function	DBOC (cm^{-1})
HF	594.96
CISD	608.70
CISDT	610.03
CASSCF(8,6)	608.28

Computed at the geometry $r_e = 0.9576$ Å, $\alpha = 104.52°$.

the correct asymptotic character of the wave function near electron–electron coalescence points, viz. Kato's cusp condition on the wave function.[73]

The terms with explicit dependence on the interelectronic distance lead to the appearance of three- and higher-electron matrix elements in the amplitude equations. Efficient evaluation of such matrix elements is possible with Kutzelnigg's idea to use the resolution of the identity coupled with partial wave analysis.[2] Thus, only two-electron matrix elements over the following operators are needed:

$$\frac{1}{r_{12}}, \quad r_{12}, \quad [\hat{\nabla}_1, r_{12}], \quad [\hat{\nabla}_2, r_{12}] \qquad (2)$$

The `LIBINT` integrals library described earlier provides the necessary functionality to compute such integrals within PSI3.[74,75]

PSI3 was the first publicly available code to implement the linear R12 version of the second-order Møller-Plesset method. At present, only energies of closed-shell species can be computed, although extensions to UHF references are underway. The two-electron integrals are computed in the AO basis and transformed to the MO basis in direct fashion using an algorithm similar to the standard direct MP2 algorithm of Frisch et al.[76] The integrals transformation can proceed in multiple passes through the AO integrals if the amount of available memory is insufficient. The integral transformation uses thread-level parallelism to take advantage of multiple processors in a shared-memory parallel machine. The largest calculations performed with the MP2-R12 code in PSI3 involved more than 1000 basis function and over 30 active orbitals.[77] Recently we computed the MP2-R12/A energy of acrylonitrile using an extended $19s14p8d6f4g4h2i/9s6p4d3f2g$ basis set[78] of 1210 functions (Table 3). The calculation required two integral transformation passes using 7 GB of memory and took 12 days on a workstation with two Intel Xeon 3.2 GHz EM64T processors.

Diagonal Born-Oppenheimer Corrections

The diagonal Born-Oppenheimer correction (DBOC; sometimes abbreviated as BODC) is the leading correction term to the Born-Oppenheimer electronic energy arising from the nuclear kinetic energy:

$$E_{\mathrm{DBOC}} = \langle \Psi_{\mathrm{e}}(\mathbf{r};\mathbf{R}) | \hat{T}_{\mathrm{nuc}} | \Psi_{\mathrm{e}}(\mathbf{r};\mathbf{R}) \rangle. \qquad (3)$$

For molecules containing light elements, the errors due to the Born-Oppenheimer approximation can be nonnegligible (e.g., >0.1 kcal mol^{-1} error in relative energies, $\sim 10^{-4}$ Å errors in bond distances, and several cm^{-1} errors in rovibrational energy levels). The DBOC must therefore be accounted for in high-accuracy thermochemistry, high-resolution rovibrational spectroscopy, etc. PSI3 can compute the DBOC using Hartree-Fock, full CI, or general RAS CI wave function, with or without orbital optimization (Table 4). To our knowledge, PSI3 was the first publicly available package able to compute the DBOC with highly correlated wave functions.

The theory behind the DBOC code in PSI3 has been described elsewhere.[79] The memory requirements of the PSI3 DBOC program scale quadratically with the number of orbital occupation strings. The limit of practical DBOC computations is therefore 20–40 thousand strings and one million determinants, depending on the available computational power. A recently implemented multithreaded out-of-core algorithm makes even larger calculations possible, especially on modern SMP machines.

Summary

PSI3 is an open-source *ab initio* quantum chemistry program package. Through robust software tools such as a TOC I/O system, libraries for easy manipulation of multi-index quantitites, and a flexible input parser, the PSI3 infrastructure offers developers a portable and convenient environment for the rapid implementation of new models. The current capabilities of the package include a wide variety of open- and closed-shell coupled cluster wave functions and properties, RAS- and full-CI wave functions, MP2-R12 models, and diagonal Born-Oppenheimer corrections for both Hartree-Fock and CI-based correlated wave functions.

Acknowledgments

The authors thank Prof. Henry F. Schaefer, in whose group the PSI package had its origin, for his support and guidance during the development of earlier versions of the program. We gratefully acknowledge additional contributions by Dr. Yukio Yamaguchi (University of Georgia), Dr. Micah Abrams (University of Central Arkansas), Nicholas Russ (Q-Chem, Inc.), John Sears (Georgia Tech), Christopher Smith (Virginia Tech), Berhane Temelso (Georgia Tech), and Mutasem Sinnokrot (Georgia Tech). We also thank Prof. Russ Pitzer of the Ohio State University for helpful discussions regarding the early history of the program package.

References

1. Kutzelnigg, W. Theor Chim Acta 1985, 68, 445.
2. Kutzelnigg, W.; Klopper, W. J Chem Phys 1991, 94, 1985.
3. Obara, S.; Saika, A. J Chem Phys 1986, 84, 3963.
4. Head-Gordon, M.; Pople, J. A. J Chem Phys 1988, 89, 5777.
5. Fermann, J. T. Efficient Implementation of Vertical Recursion Relations for the Generation of Electron Repulsion Intgerals; PhD Thesis, University of Georgia, 1996.
6. Valeev, E. F.; Allen, W. D.; Hernandez, R.; Sherrill, C. D.; Schaefer, H. F. J Chem Phys 2003, 118, 8594.

7. Klopper, W. In Encyclopedia of Computational Chemistry; Schleyer, P. v. R.; Allinger, N. L.; Clark, T.; Gasteiger, J.; Kollman, P. A.; Schaefer, H. F., III; Schreiner, P. R., Eds.; Wiley: Chichester, UK, 1998; pp. 2351–2375.
8. Stanton, J. F.; Gauss, J.; Watts, J. D.; Bartlett, R. J. J Chem Phys 1991, 94, 4334.
9. Bartlett, R. J. In Modern Electronic Structure Theory; Yarkony, D. R., Eds.; World Scientific: Singapore, 1995; Advanced Series in Physical Chemistry, Vol. 2, Chapter 16, pp. 1047–1131.
10. Lee, T. J.; Scuseria, G. E. In Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy; Langhoff, S. R., Ed.; Kluwer Academic: Dordrecht, 1995, pp. 47–108.
11. Gauss, J. Encyclopedia of Computational Chemistry; Schleyere, P. v. R.; Allinger, N. L.; Clark, T.; Gasteiger, J.; Kollman, P. A.; Schaefer, H. F., III; Schreiner, P. E., Eds.; Wiley: Chichester, UK, 1998, pp. 615–636.
12. Crawford, T. D.; Schaefer, H. F. In Reviews in Computational Chemistry; Lipkowitz, K. B.; Boyd, D. B., Eds.; VCH Publishers: New York, 2000; Vol. 14, Chapter 2, pp. 33–136.
13. Helgaker, T.; Ruden, T. A.; Jørgensen, P.; Olsen, J.; Klopper, W. J Phys Org Chem 2004, 17, 913.
14. Purvis, G. D.; Bartlett, R. J. J Chem Phys 1982, 76, 1910.
15. Rittby, M.; Bartlett, R. J. J Phys Chem 1988, 92, 3033.
16. Raghavachari, K.; Trucks, G. W.; Pople, J. A.; Head-Gordon, M. Chem Phys Lett 1989, 157, 479.
17. Bartlett, R. J.; Watts, J. D.; Kucharski, S. A.; Noga, J. Chem Phys Lett 1990, 165, 513; erratum: 1990, 167, 609.
18. Koch, H.; Christiansen, O.; Jørgensen, P.; Meras, A.; Helgaker, T. J Chem Phys 1997, 106, 1808.
19. Christiansen, O.; Koch, H.; Jørgensen, P. J Chem Phys 1995, 103, 7429.
20. Scheiner, A. C.; Scuseria, G. E.; Rice, J. E.; Lee, T. J.; Schaefer, H. F. J Chem Phys 1987, 87, 5361.
21. Salter, E. A.; Trucks, G. W.; Bartlett, R. J. J Chem Phys 1989, 90, 1752.
22. Gauss, J.; Stanton, J. F.; Bartlett, R. J. J Chem Phys 1991, 95, 2623.
23. Gauss, J.; Lauderdale, W. J.; Stanton, J. F.; Watts, J. D.; Bartlett, R. J. Chem Phys Lett 1991, 182, 207.
24. Christiansen, O.; Koch, H.; Jørgensen, P. Chem Phys Lett 1995, 243, 409.
25. Stanton, J. F.; Bartlett, R. J. J Chem Phys 1993, 98, 7029.
26. Smith, C. E.; King, R. A.; Crawford, T. D. J Chem Phys 2005, 122, 054110.
27. Stanton, J. F.; Gauss, J.; J Chem Phys 1993, 99, 8840.
28. Koch, H.; Jørgensen, P. J Chem Phys 1990, 93, 3333.
29. Koch, H.; Aa. Jensen, H. J.; Jørgensen, P.; Helgaker, T. J Chem Phys 1990, 93, 3345.
30. Kobayashi, R.; Koch, H.; Jørgensen, P. Chem Phys Lett 1994, 219, 30.
31. Pedersen, T. B.; Koch, H.; Ruud, K. J Chem Phys 1999, 110, 2883.
32. Ruud, K.; Helgaker, T. Chem Phys Lett 2002, 352, 533.
33. Ruud, K.; Stephens, P. J.; Devlin, F. J.; Taylor, P. R.; Cheeseman, J. R.; Frisch, M. J. Chem Phys Lett 2003, 373, 606.
34. Tam, M. C.; Russ, N. J.; Crawford, T. D. J Chem Phys 2004, 121, 3550.
35. Crawford, T. D.; Owens, L. S.; Tam, M. C.; Schreiner, P. R.; Koch, H. J Am Chem Soc 2005, 127, 1368.
36. Smith, C. E.; Crawford, T. D. Manuscript in preparation.
37. Crawford, T. D. Theor Chem Acc 2006, 115, 227.
38. de Meijere, A.; Khlebnikov, A. F.; Kostikov, R. R.; Kozhushkov, S. I.; Schreiner, P. R.; Wittkopp, A.; Yufit, D. S. Angew Chem Int Ed 1999, 38, 3474.
39. de Meijere, A.; Khlebnikov, A. F.; Kostikov, R. R.; Kozhushkov, S. I.; Schreiner, P. R.; Wittkopp, A.; Rinderspacher, C.; Menzel, H.; Yufit, D. S.; Howard, J. A. K. Chem Eur J 2002, 8, 828.
40. Crawford, T. D.; King, R. A. Chem Phys Lett 2002, 366, 611.
41. Russ, N. J.; Crawford, T. D. Chem Phys Lett 2004, 400, 104.
42. Sherrill, C. D.; Schaefer, H. F. In Advances in Quantum Chemistry; Löwdin, P.-O., Eds.; Academic Press: New York, 1999; Vol. 34, pp. 143–269.
43. Bochevarov, A. D.; Temelso, B.; Sherrill, C. D. J Chem Phys 2006, 125, 054109.
44. Dutta, A.; Sherrill, C. D. J Chem Phys 2003, 118, 1610.
45. Abrams, M. L.; Sherrill, C. D. J Phys Chem A 2003, 107, 5611.
46. Abrams, M. L.; Sherrill, C. D. J Chem Phys 2004, 121, 9211.
47. Sherrill, C. D.; Piecuch, P. J Chem Phys 2005, 122, 124104.
48. Leininger, M. L.; Sherrill, C. D.; Allen, W. D.; Schaefer, H. F. J Chem Phys 1998, 108, 6717.
49. Abrams, M. L.; Sherrill, C. D. J Chem Phys 2003, 118, 1604.
50. Temelso, B.; Valeev, E. F.; Sherrill, C. D. J Phys Chem A 2004, 108, 3068.
51. Krylov, A. I.; Sherrill, C. D.; Byrd, E. F. C.; Head-Gordon, M. J Chem Phys 1998, 109, 10669.
52. Krylov, A. I.; Sherrill, C. D.; Head-Gordon, M. J Chem Phys 2000, 113, 6509.
53. Krylov, A. I.; Sherrill, C. D. J Chem Phys 2002, 116, 3194.
54. Sears, J. S.; Sherrill, C. D.; Krylov, A. I. J Chem Phys 2003, 118, 9084.
55. Bochevarov, A. D.; Sherrill, C. D. J Chem Phys 2005, 122, 234110.
56. Olsen, J.; Roos, B. O.; Jørgensen, P.; Aa. Jensen, H. J. J Chem Phys 1988, 89, 2185.
57. Leininger, M. L.; Sherrill, C. D.; Allen, W. D.; Schaefer, H. F. J Comput Chem 2001, 22, 1574.
58. Shepard, R.; Shavitt, I.; Lischka, H. J Comput Chem 2002, 23, 1121.
59. Bofill, J. M.; Illas, F. Mol Phys 2003, 101, 45.
60. Shavitt, I. In Methods of Electronic Structure Theory; Schaefer, H. F., Eds.; Plenum: New York, 1977; pp. 189–275.
61. Sherrill, C. D.; Schaefer, H. F. J Phys Chem 1996, 100, 6069.
62. Krylov, A. I. Chem Phys Lett 2001, 338, 375.
63. Krylov, A. I. Chem Phys Lett 2001, 350, 522.
64. Shao, Y. H.; Head-Gordon, M.; Krylov, A. I. J Chem Phys 2003, 118, 4807.
65. Handy, N. C.; Knowles, P. J.; Somasundram, K. Theor Chim Acta 1985, 68, 87.
66. Handy, N. C. In Relativistic and Electron Correlation Effects in Molecules and Solids; Malli, G. L., Eds.; Plenum: New York, 1994; NATO ASI Series B: Physics, Vol. 318, pp. 133–160.
67. Leininger, M. L.; Allen, W. D.; Schaefer, H. F.; Sherrill, C. D. J Chem Phys 2000, 112, 9213.
68. Malmqvist, P.-A.; Rendell, A.; Roos, B. O. J Phys Chem 1990, 94, 5477.
69. Chaban, G.; Schmidt, M. W.; Gordon, M. S. Theor Chem Acc 1997, 97, 88.
70. Gdanitz, R. J. Chem Phys Lett 1993, 210, 253.
71. Noga, J.; Klopper, W.; Kutzelnigg, W. Recent Advances in Coupled-Cluster Methods; World Scientific: Singapore, 1997; p. 1.
72. Klopper, W.; Manby, F. R.; Ten-no, S.; Valeev, E. F. Int Rev Phys Chem 2006, 25, 427.
73. Kato, T. Commun Pure Appl Math 1957, 10, 151.
74. Valeev, E. F.; Schaefer, H. F. J Chem Phys 2000, 113, 3990.
75. Klopper, W.; Röhse, R. Theor Chim Acta 1992, 83, 441.
76. Frisch, M. J.; Head-Gordon, M.; Pople, J. A. Chem Phys Lett 1990, 166, 281.
77. Sinnokrot, M. O.; Valeev, E. F.; Sherrill, C. D. J Am Chem Soc 2002, 124, 10887.
78. Klopper, W.; Samson, C. C. M. J Chem Phys 2002, 116, 6397.
79. Valeev, E. F.; Sherrill, C. D. J Chem Phys 2003, 118, 3921.